UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/520,008	03/06/2000	Jong-Deok Choi	YO999-0502	7748

21254 7590 04/15/2004

MCGINN & GIBB, PLLC
8321 OLD COURTHOUSE ROAD
SUITE 200
VIENNA, VA 22182-3817

EXAMINER	
STEELMAN, MARY J	

ART UNIT	PAPER NUMBER
2122	9

DATE MAILED: 04/15/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)
	09/520,008	CHOI ET AL.
	Examiner	Art Unit
	Mary J. Steelman	2122

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *1/12/04, 2/3/04*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-69* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-69* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *03 February 2004* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☒ Other: *Accepted Drawing*.

DETAILED ACTION

1. This office action is in response to Amendment A, filed 02/03/2004. Per Applicant's request, claims 9, 26, 36, 39-40, 43-50, 53, 55, 61, 66, and 69 have been amended. Claims 1-69 are pending.

Drawings

2. Figure 11 has been accepted by the Examiner. In view of the amendments to the Specification, all other drawing objections are hereby withdrawn.

Specification

3. In view of Applicant's amendments, the objections to the Specification are hereby withdrawn.

Claim Rejections – 35 USC § 112

4. In view of the amendments to the claims, the 35 USC 112 2nd paragraph rejections are hereby withdrawn.

Double Patenting

5. In view of the Applicant's comments, the Double Patenting rejection of the former Office Action, is hereby withdrawn.

Claim Objections

6. In view of the amendment to claim 53 the prior objection to claim 53 is hereby withdrawn.

Claim 41 is objected to because the claim does not end with a '.'. Delete the ',' and add a '.'.

Claim Rejections - 35 USC § 101

7. 35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

Claims 64, 65, and 66 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter. Claims 64-66 recite "A software facility...". Software programs per se are non-statutory. This may be cured by amending the claims to recite, A software facility, tangibly embodied on a computer readable medium..." or "A software facility system..."

Claim Rejections - 35 USC § 103

8. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

9. Claims 1-33 & 35-48, 57-62, and 64-69 are rejected under 35 U.S.C. 103(a) as being unpatentable over "Deterministic Replay of Java Multithreaded Applications" by Jong-Deok

Choi, and Harini Srinivasan (August 1988), in view of "TCP/IP Illustrated, Volume 1 The Protocols", by W. Richard Stevens (1994).

Choi disclosed a deterministic replay of JAVA multithreaded applications under the name of DejaVu. Per claim 1, Choi disclosed:

-identifying an execution order of critical events of a program; (Page 2, right column, paragraph 7, "...capture all these synchronization events and the shared variable accesses in order to reproduce the exact same execution behavior...collectively refer to the synchronization events and shared variable accesses as critical events...A logical thread schedule is a sequence of intervals of critical events...")

-generating groups of critical events of said program, wherein for each group, critical events belonging to said group belong to a common execution thread; (Page 4, left column, paragraphs 3 & 4, "The logical thread schedule of an execution instance...is an ordered set of critical events intervals...a set of maximally consecutive critical events of a thread. Formally, a logical schedule interval is a (temporally well ordered) non-empty set of critical events with the following properties: 1. all critical events of the logical schedule interval belong to the same thread...")

-generating, for each execution thread, a logical thread schedule that identifies a sequence of said groups so as to allow deterministically replaying non-deterministic bytes (Page 4, left column, paragraph 5, "...given any two critical events...of the logical schedule interval, all critical events of the thread that happened between (two points) also belong to this logical schedule interval...")

Although Choi did not provide specific information on networking features, Choi did allow for the possibility of the DejaVu system to operate in a networked environment. At

several locations in the article Choi noted that "it can be used on a multiprocessor system as well" (page 1, right column, 4th paragraph). The presented claims read exactly on DejaVu with the addition of networking features such as socket connections and datagram messages included. Choi did not address replaying a non-deterministic arrival of stream socket connection requests, a non-deterministic number of bytes received during message reads, a non-deterministic binding of stream sockets to local ports, and a non-deterministic arrival of datagram messages. However, networking protocols are well known and described by Stevens (Chapter 2: TCP/IP, PPP, datagrams, chapter 11: UDP, out of order delivery, reassembly, header data, path discovery, chapter 12: unicasting and multicasting, chapter 17: TCP headers, sequencing number, source / destination port numbers, packet numbers, order events, chapter 18: socket connections, network ids, synchronizing, chapter 25: identifiers / tables (logging))

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention, to have modified Choi's disclosed invention, a modified virtual machine that records and replays a program's execution, deterministically enforcing logical thread schedules by including networking features as disclosed by Stevens because Choi did disclose that his DejaVu implementation could be used on a multiprocessor system, but failed to disclose common network features. Stream socket connection requests, message reads, stream sockets to local ports, and datagram messages are features that allow for communication between networked computers. They are well known to persons of ordinary skill in the art and are disclosed in the Stevens textbook.

Regarding claims 2-10, 17-33, and 39, Choi disclosed:

Per claim 2: (Page 9, right column, 5th paragraph, "We have implemented the record/replay mechanism discussed in the previous sections by modifying the ...Virtual Machine...")

Per claim 3: (Page 1, right column, end of 4th paragraph, "Another advantage of DejaVu is that it can be used on a multiprocessor system...")

Per claim 4: (Page 2, right column, middle of 7th paragraph, "It is therefore imperative, in a record/replay tool, to capture all these synchronization events and the shared variable accesses in order to reproduce the exact same execution behavior o the program.")

Per claim 5: (Page 9, right column, 5th paragraph, "...record/replay mechanism...modifying the...Virtual Machine...is capable of deterministic replay of JAVA multithreaded programs...")

Per claim 6: (Page 9, right column, 5th paragraph, "We considered modifying the application bytecode...but decided against it...")

Per claim 7: (Page 1, right column, 4th paragraph & 6th paragraph, "Replaying a multithreaded program on a uniprocessor system can be achieved by first capturing the thread schedule information during one execution of the program, and then enforcing the exact same schedule when replaying...")

Per claim 8: (Page 11, right column, 3rd paragraph, "Also, our approach works on multiprocessor systems..." and page 9, left column, 2nd paragraph, "...each thread executes and reproduces the same execution behavior using this ordered list of schedule intervals")

Per claim 9: (Page 2, right column, paragraph 7, "A logical thread schedule is a sequence of intervals of critical events, wherein each interval corresponds to the critical and non-critical events executing consecutively in a specific thread." and page 4, right column, paragraphs 3 & 5, "Although the global clock and a thread's local clock...")

Per claim 10: (Page 4, left column, paragraph 8, - right column, paragraph 1, "We capture this unique logical thread schedule during the record phase, and enforce it during the replay phases to reproduce the same execution behavior.")

Per claim 17: (Page 4, left column, paragraph 8-right column, paragraph 1, "We capture this unique logical thread schedule during the record phase, and enforce it during the replay phases to reproduce the same execution behavior.")

Per claim 18: (Page 9, left column, paragraph 2, "...each thread executes and reproduces the same execution behavior using this ordered list of schedule intervals")

Per claim 19: (Page 2, right column, paragraph 1, "We collectively refer to all the physical thread schedules in an equivalence class as a logical thread schedule.")

Per claim 20: (Page 2, right column, paragraph 7, "We collectively refer to the synchronization events and shared variable accesses as critical events." Also page 2, right column, paragraph 3, "monitorenter, monitorexit that mark the begin and end, respectively, of a critical section...")

Per claim 21: (Page 2, right column, paragraph 4, "wait, notify/notifyAll that can be used to coordinate the execution order of multiple threads...")

Per claim 22: (Page 2, right column, paragraph 7, "A logical thread schedule is a sequence of intervals of critical events...)

Per claim 23: (Page 4, left column, paragraphs 3-6, "Each logical schedule interval is a set of maximally consecutive critical events of a thread. Formally a logical schedule interval is a (temporally well ordered) non-empty set of critical events with the following properties: 1. all critical events of the logical schedule interval belong to the same thread; 2. given any two

critical events...all critical events of the thread that happened between (two points) also belong to this logical schedule interval; and 3. no two adjacent intervals belong to the same thread.")

Per claim 24: (Page 4, left column, paragraph 1, "...trace the first access to f, which is a read, and the last access to g...")

Per claims 25, 26, 27, 28 & 30: (Page 4, right column paragraphs 2 & 3, "Each schedule interval, LSI, is an ordered set of critical events, and can be represented by its first and last critical events as follows...We use the global clock that ticks at each execution of a critical event to uniquely identify each critical event...While running, each thread captures the *FirstCriticalEvent* and *LastCriticalEvent*... ")

Per claim 29: (Page 4, right column, paragraph 5, "...global clock and a thread's local clock start with the same time value, the local clock stays behind the global clock when a different thread executes a critical event...")

Per claim 31: (Page 4, right column, paragraph 9, "The thread executes the critical event, and increments the global clock, both as one atomic operation...")

Per claim 32: (Page 5, left column, paragraphs 2 & 3 – right column, paragraphs 1 & 2, "Since multiple threads execute critical events and update the same global clock, the following three events must be executed as a single atomic action...1. AssignGlobalClock – assigning the global clock value to the critical event; 2. UpdateGlobalClock – incrementing the global clock...3. CriticalEvent – execution of a critical event...)

Per claim 33: (Page 9, left column, paragraph 3-8 & right column, paragraph 1.)

Per claim 39: (Page 4, left column, paragraph 8, through right column paragraph 1.)

Regarding claims 11-16, 35-38, 40-48, 57-62, and 64-69:

Choi disclosed a deterministic replay of JAVA multithreaded applications under the name of DejaVu. Although Choi did not provide specific information on networking features, Choi did allow for the possibility of the DejaVu system to operate in a networked environment. At several locations in the article Choi noted that "it can be used on a multiprocessor system as well" (page 1, right column, 4th paragraph). The presented claims read exactly on DejaVu with the addition of networking features such as socket connections and datagram messages included.

Choi did not address PPP (Point-to-Point Protocols), limitations found in claims 11, 12, and 16. However, Stevens addressed the PPP in chapter 2, pages 26-27 & 31-32.

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention, to have combined the disclosed art of Choi, a deterministic replay of JAVA multithreaded applications, by combining the features found in PPP networking, as addressed by Stevens, because Choi did disclose that his invention could be used on a multiprocessor system and by including the use of PPP, a well known faster network traffic medium, as noted by Stevens, processor to processor communications can realize an improvement.

Choi disclosed a deterministic replay of JAVA multithreaded applications under the name of DejaVu. Although Choi did not provide specific information on networking features, Choi did allow for the possibility of the DejaVu system to operate in a networked environment. At several locations in the article Choi noted that "it can be used on a multiprocessor system as well" (page 1, right column, 4th paragraph). The presented claims read exactly on DejaVu with the addition of networking features such as socket connections and datagram messages included.

Choi did not address out-of-order packets, reassembly after fragmentation, identification, header data, and route identification, limitations of claims 38, 41, 42, 45, 46, 57-61, 64-69.

However, Stevens addressed these features in chapter 11: (discover path / page 153-157, header data / 148-157, identification / pages 148-150, reassembling after fragmentation / pages 148-149, out-of-order connections / pages 148-149, user datagram protocol sockets / pages 159-160, source IP and port number / page 144, handling datagram size / pages 159-160). While Stevens does not 'count' the "number of packets delivered", the packets are certainly accounted for and it is known whether all packet fragments have arrived and if they are successfully assembled into the datagram. See Stevens p. 148, section 11.5 – p. 167, 11.13.

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention, to have combined the disclosed art of Choi, a deterministic replay of JAVA multithreaded applications, with common networking features including "UDP, a datagram oriented transport layer protocol" (page 143), in support of execution replay with respect to said datagram socket Application Programming Interface, because Choi did disclose that his invention could be used on a multiprocessor system and datagram protocols, identifications of paths, sources and ports, as noted by Stevens, are commonly used in sending / receiving and verifying processor to processor communications found in many multiprocessor systems.

Choi disclosed a deterministic replay of JAVA multithreaded applications under the name of DejaVu. Although Choi did not provide specific information on networking features, Choi did allow for the possibility of the DejaVu system to operate in a networked environment. At several locations in the article Choi noted that "it can be used on a multiprocessor system as

well" (page 1, right column, 4th paragraph). The presented claims read exactly on DejaVu with the addition of networking features such as socket connections and datagram messages included.

Choi did not address unicasting and multicasting, limitations of claims 14, 15, 40, and 62.

However, Stevens addressed these features in chapter 12: ("three kinds of IP addresses: unicast, broadcast, and multicast (threads in parallel)" / page 169).

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention, to have combined the disclosed art of Choi, a deterministic replay of JAVA multithreaded applications, by including options for unicasting or multicasting, in referencing to sending messages, as noted by Stevens, because it allows the sender to efficiently specify one recipient or multiple recipients as needed when communicating in a multiprocessor system.

Choi disclosed a deterministic replay of JAVA multithreaded applications under the name of DejaVu. Although Choi did not provide specific information on networking features, Choi did allow for the possibility of the DejaVu system to operate in a networked environment. At several locations in the article Choi noted, "it can be used on a multiprocessor system as well" (page 1, right column, 4th paragraph). The presented claims read exactly on DejaVu with the addition of networking features such as socket connections and datagram messages included.

Choi did not address TCP headers, sequence numbers, source and destination port numbers, packet numbers and the order of events, limitations of claims 37, 60, 61, and 64-69.

However, Stevens addressed these features in chapter 17: (order events within thread / page 227, TCP headers, sequence numbers, source and destination port numbers, number of packets / pages 225-228) .

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention, to have combined the disclosed art of Choi, a deterministic replay of JAVA multithreaded applications, with features addressed by Stevens, regarding Transmission Control Protocol, because "TCP provides a connection-oriented, reliable, byte stream service" (page 223) useful when communicating in a multiprocessor system.

Choi disclosed a deterministic replay of JAVA multithreaded applications under the name of DejaVu. Although Choi did not provide specific information on networking features, Choi did allow for the possibility of the DejaVu system to operate in a networked environment. At several locations in the article Choi noted, "it can be used on a multiprocessor system as well" (page 1, right column, 4th paragraph). The presented claims read exactly on DejaVu with the addition of networking features such as socket connections and datagram messages included.

Choi did not address connections, network id, and synchronizing, limitations of claims 12, 13, 35-38, 41-44, and 46-48. However, Stevens addressed these features in chapter 18: (connection establishment & termination protocol / page 229-234, synchronizing (SYN) / page 231, network id / page 237, bind socket to port & unique identity / page 260, (uniquely identified by a 4-tuple), re-establish connection / page 230, identifying a connection request / page 240) .

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention, to have combined the disclosed art of Choi, a deterministic replay of JAVA multithreaded applications, with features addressed by Stevens, regarding the connection establishment and termination of TCP, because TCP is a useful technique for exchanging data among networked processors, a connection-oriented, reliable, byte stream service.

Choi disclosed a deterministic replay of JAVA multithreaded applications under the name of DejaVu. Although Choi did not provide specific information on networking features, Choi did allow for the possibility of the DejaVu system to operate in a networked environment. At several locations in the article Choi noted that "it can be used on a multiprocessor system as well" (page 1, right column, 4th paragraph). The presented claims read exactly on DejaVu with the addition of networking features such as socket connections and datagram messages included.

Choi did not address identifiers and logging information in tables, limitations of claims 43 & 44. However, Stevens addressed these features in chapter 25: (variable identifiers and tables, logging / page 388).

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention, to have combined the disclosed art of Choi, a deterministic replay of JAVA multithreaded applications, by combining the common networking features including "UDP, a datagram oriented transport layer protocol" (page 143), because Choi did disclose that his invention could be used on a multiprocessor system and the use of identifiers, tables, and logging, as noted by Stevens, are useful for network management in processor to processor communications.

10. Claims 34 and 49-56 are rejected under 35 U.S.C. 103(a) as being unpatentable over "Deterministic Replay of Java Multithreaded Applications" by Jong-Deok Choi, and Harini Srinivasan (August 1988), in view of "TCP/IP Illustrated, Volume 1 The Protocols", by W.

Richard Stevens (1994), and further in view of "The JAVA Developers Almanac, The JAVA Series from the Source" by Patrick Chan (1998).

Choi disclosed a deterministic replay of JAVA multithreaded applications under the name of DejaVu. Although Choi did not provide specific information on networking features, Choi did allow for the possibility of the DejaVu system to operate in a networked environment. At several locations in the article Choi noted that "it can be used on a multiprocessor system as well" (page 1, right column, 4th paragraph). The presented claims read exactly on DejaVu with the addition of networking features such as socket connections and datagram messages included.

Choi did not address socket connections, TCP/IP, PPP and other networking features. However, these are well known, and addressed by Stevens. Choi and Stevens, combined, failed to address the JAVA language specification. However, Chan, included information on JAVA packages, classes, and methods included as a part of the JDK, (java.io, java.net, java.rmi*) (pages 63-80) that can be used to enable byte code to work with socket APIs, and the JAVA Native Interface (JNI) (pages 794-799), a native programming interface for JAVA that is a part of the JDK, used to write native methods (map socket calls in a virtual machine). Chan referenced (pages 63-66), java.io classes for various read() methods available, (page 74) SocketPermission and BindException in the java.net package, referenced by claim limitations of claims 49, 54, 55, and 56. Chan included information on JAVA Native Interface, JNI at pages 794-799, referenced by claim limitations of claim 34.

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention, to have modified Choi's DejaVu technique, which could be used on a multiprocessor system, to include socket API methods as disclosed by Stevens and JAVA

methods, including native interface methods, to address networking features of virtual machines and byte codes, because these are common communication methods for a virtual machine networked environment that enable interprocess communications, and they are defined in the Java Developers Kit as a part of the JAVA language, as addressed by Chan.

Per claim 50: (See limitations addressed in claims 33 and 49.)
Per claims 51 and 52, Stevens disclosed (page 505, Appendix C, The Sock Program, first paragraph) "specify the size of each read".

Therefore, it would have been obvious to one of ordinary skill in the art, at the time of the invention, to have modified Choi's DejaVu technique, which could be used on a multiprocessor system, to include socket API methods as disclosed by Stevens, because the Choi/Srinivasan reference clearly allows for a multiprocessor system and the use of socket protocols in multiprocessors is well known for networked communication.

Per claim 53, Stevens disclosed (chapter 11, page 149), "The information maintained in the IP header for fragmentation and reassembly provides enough information to do this."

Therefore, it would have been obvious to one of ordinary skill in the art, at the time of the invention, to have modified Choi's DejaVu technique, which could be used on a multiprocessor system, to include IP header information as disclosed by Stevens, because the Choi/Srinivasan reference clearly allows for a multiprocessor system and the use of TCP/IP protocols in multiprocessors, is well known for networked communication.

Response to Arguments

11. Applicant's arguments filed 02/03/2003, Amendment A, have been fully considered but they are not persuasive.

(A) Applicant, in substance, has argued the following:

As Applicant has pointed out on page 25, paragraphs 1-6, of Amendment A, there is no motivation to combine the Choi/Srinivasan Article and the Stevens Book. The references are unrelated as Choi/Srinivasan is directed to a record/replay tool, whereas Stevens is merely directed to a TCP/IP protocols.

Examiner's Response:

In response to applicant's argument that there is no suggestion to combine the references, the examiner recognizes that obviousness can only be established by combining or modifying the teachings of the prior art to produce the claimed invention where there is some teaching, suggestion, or motivation to do so found either in the references themselves or in the knowledge generally available to one of ordinary skill in the art. See *In re Fine*, 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988)and *In re Jones*, 958 F.2d 347, 21 USPQ2d 1941 (Fed. Cir. 1992). In this case, the Choi/Srinivasan reference disclosed that their JAVA debugger "...can be used on a multiprocessor system..." (page 1, right column, 4th paragraph). A multiprocessor system, which commonly is deployed on multiple machines, requires I/O communication protocol. The Stevens reference supplies well known TCP/IP and User Datagram communication protocols. Thus there exists motivation for combining the references.

(B) Applicant, in substance, has argued the following:

As Applicant has pointed out on page 26, paragraphs 1-3, of Amendment A, neither the Choi/Srinivasan nor the Stevens reference teach:

-generating for each execution thread, a logical thread schedule that identifies a sequence of said groups so as to allow deterministically replaying a non-deterministic arrival of stream socket connection requests, a non-deterministic number of bytes received during message reads, a non-deterministic binding of stream sockets to local ports, and a non-deterministic arrival of datagram messages"

-determining a non-deterministic number of packets delivered during different executions of the same program, for supporting an execution replay with respect to said datagram socket Application Programming Interface (API)."

Examiner's Response:

Per claim 1: -generating, for each execution thread, a logical thread schedule that identifies a sequence of said groups so as to allow deterministically replaying non-deterministic bytes (Page 4, left column, paragraph 5, "...given any two critical events...of the logical schedule interval, all critical events of the thread that happened between (two points) also belong to this logical schedule interval...")

Per claims 61, 66 and 69: determining a non-deterministic number of packets delivered during different executions of the same program, for supporting an execution replay with respect to said datagram socket Application Programming Interface (API). This is a newly added

limitation. Although Stevens doesn't 'count' packets delivered, he 'accounts' for each packet's successful arrival, at which time they are reassembled into the datagram. See Stevens p. 148-167 and p. 496: socket debug.

(C) Applicant, in substance, has argued the following:

As Applicant has pointed out on page 27, paragraphs 1-6, of Amendment A, there is no motivation to combine the Choi/Srinivasan Article, Stevens, and the Chan Book. The references are unrelated as Choi/Srinivasan is directed to a record/replay tool, Stevens is related to TCP/IP protocol, whereas Chan discloses JAVA packages, classes and methods..

Examiner's Response:

In response to applicant's argument that there is no suggestion to combine the references, the examiner recognizes that obviousness can only be established by combining or modifying the teachings of the prior art to produce the claimed invention where there is some teaching, suggestion, or motivation to do so found either in the references themselves or in the knowledge generally available to one of ordinary skill in the art. See *In re Fine*, 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988)and *In re Jones*, 958 F.2d 347, 21 USPQ2d 1941 (Fed. Cir. 1992). In this case, the Choi/Srinivasan reference disclosed that their JAVA debugger "it can be used on a multiprocessor system..." (page 1, right column, 4th paragraph). A multiprocessor system, which commonly is deployed on multiple machines, requires I/O communication protocol. The Stevens reference supplies well known TCP/IP communication protocols. The Chan reference supplies communication protocols specific to the JAVA language. Thus the combination is warranted.

(D) Applicant, in substance, has argued the following:

As Applicant has pointed out on page 28, 4th paragraph, of Amendment A, "multiprocessor applications do not imply that they always have network I/O operations in them."

Examiner's Response:

The Choi/Srinivasan reference disclosed, "it can be used on a multiprocessor system..." (page 1, right column, 4th paragraph). While multiprocessor systems may not always be networked, they commonly are networked, thus requiring some type of I/O communication between machines.

Examiner maintains the rejections of claims 1-69.

Conclusion

12. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

13. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Mary Steelman, whose telephone number is (703) 305-4564. The examiner can normally be reached Monday through Thursday, from 7:00 A.M. to 5:30 P.M. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Dam can be reached on (703) 305-4552.

The fax phone number is (703) 872-9306. Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is (703) 305-3900.

Mary Steelman

03/29/2004

ANTONY NGUYEN-BA
PRIMARY EXAMINER